REMARKS

Claims 1 – 2 and 4 – 27 remain in the application. Claims 11 – 27 have been allowed. Claims 3 – 10 are objected to from depending from a rejected base claim. Claim 3 is canceled and rewritten as new claim 28. Claims 4, 5, 8 and 10 are amended herein. Claims 1 and 2 stand rejected. No new matter is added.

Claims 3 – 10 are objected to for depending from a rejected independent claim. Claims 5, 8 and 10 are also objected to for formal reasons. Responsive thereto, claims 4, 5, 8 and 10 are amended herein and claim 3 is canceled and rewritten as new claim 28. Accordingly, claims 28 and 4 – 10 are believed to be allowable. Reconsideration and withdrawal of the objection to claims 4 – 10 and allowance of claims 28 and 4 – 10 is respectfully solicited.

The Examiner asserts that claims 1 and 2 are unpatentable over Hunt et al. (U.S. Patent No. 6,629,123) in view of Ibe et al. (U.S. Patent No. 6,437,804) under 35 U.S.C. §103(a). The rejection is respectfully traversed.

The Examiner asserts that "Hunt discloses a task management method for determining optimal placement of task components," substantially as recited in 4 of the 5 steps of claim 1. The Examiner asserts that, essentially, the first recited step (a) of "generating a communication graph representative of a task" is disclosed by Hunt et al. at "col. 23 lines 13-23, 'the application units and inter-unit communication form a commodity flow network', wherein the application units are components of a task or an application program, and the inter-unit communication provides information pertaining to the weighting of edges... ." The Examiner proceeds to the third step (c), asserting that "identifying high communication edges within said communication graph, said high communication edges having a weight indicating a communication level exceeding the

communication level for a selected terminal node" is described by: "For each application unit that must reside on the client-for instance, because it directly accesses GUI functions-create an edge with **infinite weight from the source to the application unit**. For each application unit that must reside on the server-because it directly accesses storage-create an edge with infinite weight between **the sink and the application unit**" in Hunt et al. at col. 24, lines 8 – 28 (emphasis added). The Examiner asserts that step (d) of "determining a min cut for the communication graph, high communication edges being excluded min cut solutions" is disclosed in Hunt et al. at "col. 24 lines 8-28, 'the algorithm to map a client-server distributed partitioning problem onto the MIN-CUT problem is as follows', wherein the algorithm for determining the minimum cut of the graph is disclosed". Finally, the Examiner asserts that the fourth step (e) of "placing task components responsive to said min cut determined for each independent net" (claim 1, lines 6 – 7), the Examiner asserts is "placing task components responsive to said min cut determined for the communication graph" that the Examiner asserts is disclosed by Hunt et al. at "col. 23 lines 13-23, 'After all data has been gathered, it is the optimization algorithm that decides where application units will be placed on the network', wherein the optimization algorithm is a minimum cut algorithm and finds the paths of minimal communication costs".

Hunt et al. teaches "An automatic distributed partitioning system (ADPS) intercepts function calls to unit activation functions that dynamically create application units, such as a component instantiation function." *See*, Hunt et al. Abstract. A "distribution optimization algorithm accepts a model of the decision problem and ... decides where application units will be placed in the network." *Id*, col. 23, lines 13 – 17. "[T]he application units and inter-unit communication form a commodity flow network. After this mapping, known graph-cutting algorithms can be used for automatic distributed partitioning." *Id*, lines 20 – 23. Hunt et al. also teaches that "the minimum cut contains edges with the smallest weights (capacities), those edges represent the line of minimum communication between the client and server." *Id*, col. 24, lines 25 – 28.

Regarding the Examiner's assertion that Hunt et al. teaches the third step (c); Hunt et al. at col. 24, lines 8 – 28 teaches something quite different than "identifying high communication edges within said communication graph, said high communication edges having a weight indicating a communication level exceeding the communication level for a selected terminal node" as claim 1 recites. In particular assigning an infinite weight to edges between "each application unit that must reside on the client-for instance, because it directly accesses GUI functions ... (and) each application unit that must reside on the server-because it directly accesses storage" *ignores communications* levels between nodes. Instead, weights are assigned based on placement. In point of fact, one such application unit residing on the Hunt et al. client may communicate with another such application unit residing on the Hunt et al. server much more frequently than either communicates with the GUI or storage. Clearly, Hunt et al. (and Ibe et al.) ignores this and neither suggests or discloses step (c) as recited in claim 1.

The Examiner looks to Ibe et al. to disclose step (b) of assigning terminal nodes to said communication graph to be described by "'Shaded nodes 3, 6, and 10 are anchor nodes', wherein an anchor node is analogous to the claimed terminal nodes in both form and function" at col. 5 line 64 – col. 6 line 3.

However, Ibe et al. teaches modeling a network as a graph, partitioning the graph, "assigning a **weight to each node** in the graph, and... balancing partitions as a function of the **weight of each node** in a respective partition." *See*, col. 2, lines 37 – 43 (emphasis added). Ibe et al. types edges or links as normal, strong and weak (which are not included in any partition). *See*, col. 5, lines 47 – 51. Aside from the apparent differences of weighting nodes instead of edges and typing edges into 2 types, Ibe et al. is distinguished in that a "node on which a control agent is attached is defined as an 'anchor node.'" Col. 6, lines 24 – 25. Further, "an automated system has the advantage that it can have a built-in mechanism that allows the network control agents to monitor one another's status." Col. 3, lines 11 – 13. The present specification clearly recites that

"(t)erminal nodes representative of the multiple computers are attached to the communication graph." Page 5, lines 13 – 14. Accordingly, while the terminal nodes may include a control agent, a control agent is not necessarily a terminal node and vice versa. Therefore, Ibe et al. anchor nodes are quite different from terminal nodes in both form and function and, Hunt et al. in combination with Ibe et al. does not result in the present invention as recited in claims 1 and 2. Reconsideration and withdrawal of the rejection of claims 1 and 2 over Hunt et al. in view of Ibe et al. under 35 U.S.C. §103(a) is respectfully solicited.

The applicants have considered the other references cited but not relied upon in the rejection and find them to be no more relevant than the references upon which this rejection is based.

The applicants thank the Examiner for efforts, both past and present, in examining the application. Believing the application to be in condition for allowance, both for the amendment to the claims and for the reasons set forth above, the applicants respectfully request that the Examiner reconsider and withdraw the objection to claims 4 – 11 and the rejection of claims 1 and 2 under 35 U.S.C. §103(a) and allow the application to issue.

Should the Examiner believe anything further may be required, the Examiner is requested to contact the undersigned attorney at the local telephone number listed below for a telephonic or personal interview to discuss any other changes.

Please charge any deficiencies in fees and credit any overpayment of fees to IBM Corporation Deposit Account No. 50-0510 and advise us accordingly.

Respectfully Submitted,

April 28, 2004
(Date)

Charles W. Peterson, Jr.
Registration No. 34,406

Customer No. 33233
Law Office of Charles W. Peterson, Jr.
P.O. Box 710627
Oak Hill, VA 20171
Telephone: (703) 481-0532
Facsimile: (703) 481-0585